Automatic human trajectory destination prediction from video

Palwasha Afsar[a,*], Paulo Cortez[a], Henrique Santos[a]

[a] *ALGORITMI Research Centre, Department of Information Systems, University of Minho, 4800-058 Guimarães, Portugal*

Abstract

This paper presents an intelligent human trajectory destination detection system from video. The system assumes a passive collection of video from a wide scene used by humans in their daily motion activities such as walking towards a door. The proposed system includes three main modules, namely human blob detection, star skeleton detection and destination area prediction, and it works directly with raw video, producing motion features for destination prediction system, such as position, velocity and acceleration from detected human skeletons, resulting in several input features that are used to train a machine learning classifier. We adopted a university campus exterior scene for the experimental study, which includes 348 pedestrian trajectories from 171 videos and five destination areas: A, B, C, D and E. A total of six data processing combinations and four machine learning classifiers were compared, under a realistic growing window evaluation. Overall, high quality results were achieved by the best model, which uses 37 skeleton motion inputs, undersampling on training data and a random forest. The global discrimination, in terms of area of the receiver operating characteristic curve is around 87%. Furthermore, the best model can predict in advance the five destination classes, obtaining a very good ahead discrimination for classes A, B, C and D, and a reasonable ahead discrimination for class E.

Keywords: Video, Computer Vision, Machine Learning, Multi-class classification

1. Introduction

Due to advances in information technology (e.g., big data, Internet of things), human activities are increasingly being automatically recorded in a digital form. In particular, digital cameras are widely used in indoor and outdoor locations, leading to a growing demand of intelligent systems to analyze human behavior from video data (Mabrouk and Zagrouba, 2017). Often, these systems address one or more of these three intelligent video analysis aspects: abnormal event detection (e.g., intrusion, loitering, accidents) (Mabrouk and Zagrouba, 2017; Cermeño et al., 2018); person identification and tracking (e.g., gender detection, person entering into a commercial store, person walking path) (Duque et al., 2007; Afsar et al., 2015b; Cortez et al., 2016); and activity modeling (e.g., cooking, running, using a smartphone) (Afsar et al., 2015a). There are several real-world application domains that

*Corresponding author

Email addresses: palo_afsar77@yahoo.com (Palwasha Afsar), pcortez@dsi.uminho.pt (Paulo Cortez), hsantos@dsi.uminho.pt (Henrique Santos)

can potentially benefit from such video analysis, such as: security, health and well-being (Mabrouk and Zagrouba, 2017); gamming (Afsar et al., 2015b); and marketing and retail management (Cortez et al., 2016).

The problem of tracking and estimating human body keypoints in complex, multi person videos has been done by (Girdhar et al., 2017). Their approach seems to be very lightweight yet effective utilizing all of the latest advancements of human detection and video understanding. The algorithm has been tested on PoseTrack database using Convolutional Neural Net (CNN) and 3D mask R-CNN, which obtains state of the art performance. Zhou et al. (2016) also utilized deep fully convolutional network for the estimation of human pose from monocular video. Empirical evaluation on the Human3.6M dataset shows that the proposed approache achieves greater 3D pose estimation accuracy over state-of-the-art baselines. In order to estimate human pose in unconstrained videos, Zhang and Shah (2015) deployed tree-based optimization scheme. The proposed approach is based on abstraction and association to enforce the intra and inter-frame body part constraints respectively without introducing extra computational complexity. The algorithm is tested on three publically available datasets with improved performance.

This work addresses pedestrian trajectory destination estimation, which can be related with all three video analysis aspects. The final location of a walking person is a relevant component of pedestrian tracking systems, which can also be used to generate data for the prediction models. Moreover, several pedestrian destination locations can be associated with activities (e.g., cashier machine for payment, automatic teller machine for cash withdrawal) or abnormal events (e.g., violation of a restricted area, person crossing a railway).

Given the importance of this topic, several works have been proposed for pedestrian trajectory prediction. For instance, Lin et al. (2016) utilized a novel localization method (LNM) based on Markov-chain prediction and neighbor relative RSS (NRRSS), which mainly works on finger-print technology and Markov chain models for providing accurate location results with low calibration requirements. The proposed system can provide robust and accurate location information against device heterogeneity and environmental dynamics. To solve the problem of occluded objects or objects with similar appearances, Sadeghian et al. (2017) used Recurrent Neural Networks (RNN) that can reason on multiple cues over a temporal window using online information without the need to see future frames. The algorithm can track multiple targets using their motion, appearance and interactions. Following the recent success of Recurrent Neural Network (RNN) models for sequence prediction tasks, Alahi et al. (2016) worked on Long Short Term Memory (LSTM, a variant of RNN) to correctly predict the future path and destinations of pedestrians. Such systems can be help for autonomous vehicle navigating to foresee the future positions of pedestrians and accordingly adjust its path to avoid collisions. The Robicquet et al. (2016) work is based on a versatile dataset that not only includes pedestrians but also bicyclists, skateboarders, buses, golf carts sharing the same space. Their research is focused on target trajectory forecasting and Multi-Target Tracking (MTT), where a learnt model is utilized for enhancing tracking results. While current trajectory prediction systems are useful in certain applications, they fail in describing the position and behavior of moving objects in a network constrained environment. To solve this problem, Qiao et al. (2015) proposed Hidden Markov model-based Trajectory Prediction (HMTP), that captures the required parameters from real-world in terms of objects at varying speed. The proposed system is able to predict continuous path

of moving objects rather than slices of trajectory patterns.

Luber et al. (2010) combined a tracker with a dynamic pedestrian model for more realistic human motion predictions, reaching interesting performances when collecting data using small scenes and laser scanners. Yamaguchi et al. (2011) proposed social force models, which consider interactions between individuals, in order to predict pedestrian destination. Kratz and Nishino (2012) used hidden markov model trained on spatio-temporal motion patterns to predict the next local spatio-temporal motion pattern, aiming to track individuals in crowded complex scenes. Kim et al. (2015) approached next pedestrian trajectory positions using human velocity features and scene obstacles computed from video frames. More recently, Fernando et al. (2017) used deep learning neural network that combined both individual and neighborhood data to predict future pedestrian motion. In the same year, Lee et al. (2017) proposed a deep stochastic framework for predicting vehicle and pedestrian motion trajectories, using historical features related with both individual and neighborhood motion.

Most of the previous works aim to predict continuous pedestrian trajectory, either used to determine the next probable position of a person (e.g., (Kratz and Nishino, 2012; Kim et al., 2015)) or predict its whole trajectory in terms of its 2D path (e.g., (Fernando et al., 2017; Lee et al., 2017)), thus adopting regression tasks. In this paper, and similarly to the work of (Yamaguchi et al., 2011), we follow a computationally simpler approach where we directly predict the final pedestrian destination in the scene and thus it can be also viewed as a modeling task, if some destination areas are highly correlated with activities. For instance, some destination areas could correspond to activities such as entering a building or executing a payment near a cashier machine. We assume that a recorded scene contains a set of few destinations, each defined in terms of a static region of interest (e.g., door, leisure space or restricted area). Depending on the real-world application, it is reasonable to assume that such destinations can be previously defined by the domain user for a specific video scene (e.g., exit door, entrance of commercial store, cashier machine) or automatically collected by analyzing historical human scene entry and exit points. In particular, we assume a passive person detection system, where a digital camera is installed in such a way that it can capture a wide area used by pedestrians. We further note that Yamaguchi et al. (2011) used clean preprocessed video datasets, thus working directly with tracking sequences that were complemented by manual annotations of scene obstacles (similarly to (Kim et al., 2015)). Moreover, Yamaguchi et al. (2011) tested only one classifier, Support Vector Machine (SVM). In contrast, our proposed system:

i) Works directly with raw video data, recorded on a realistic and semi-constrained environment (Afsar et al., 2017), related with a university campus outdoor scene with five destination regions and that included glass reflection, different weather conditions (e.g., rain and wind) and varying illumination conditions;

ii) Extracts motion features for pedestrian destination prediction (position, velocity and acceleration) from human skeletons that are automatically identified from raw images, which includes the human body center of mass and head, hand and leg positions, and that proved useful for detecting human activities (e.g., walking, sitting) (Afsar et al., 2015a);

iii) Its fully automatic, thus it does not require manual annotations of scene obstacles

or events (as in (Yamaguchi et al., 2011; Kim et al., 2015)), only a (optional) prior definition of static destination regions of interest;

iii) Compares four machine learning classifiers (Multinomial Logistic Regression, Multilayer Perceptron, SVM and Random Forest), under distinct feature extraction and balanced training setups; and

iv) Is evaluated using a more realistic growing window procedure, in terms of ahead discrimination capability, measured in terms of the area of the Receiver Operating Characteristic (ROC) curve that is obtained through time, before reaching the destination.

The paper is organized as follows. Section 2 presents the collected video dataset, the proposed system framework and evaluation procedure. Next, Section 3 describes the conducted experiments and obtained results. Finally, Section 4 draws the main conclusions and suggestions of future work.

2. Material and Methods

2.1. Video Dataset

For this work, a video dataset was recorded at university of Minho. This dataset was used for different tasks (e.g., detection of walking and sitting actions) in our previous works (Afsar et al., 2015a, 2017) The dataset was recorded for a total of 7 days during working hours (from 8h00 to 19h00). For recording the dataset, two cameras HIK Vision and IR Network were installed, one inside the campus and one outside the campus, as shown in Figure 1. We adopted a passive collection of videos, where the environment was not controlled and thus all actions recorded are related with real-life human actions (from students, researchers and other university staff). Distinct actions were recorded, mostly walking or running, but also group interactions, talking on mobile, standing and shaking hands. To save disk space, the videos were only recorded when movement was detected. As such, the dataset is comprised of hundreds of small videos (with a few seconds to a few minutes each) that correspond to 32GB.

Figure 1: Installation setup of passive cameras for capturing indoor (left) and output (right) campus environments, source: (Afsar et al., 2015a)

The videos were recorded for indoor and outdoor scenes, as depicted in the examples of Figure 2. Since we adopted a real environment, the recorded data includes several restrictions that pose challenges: the cameras were set in front of a glass window (thus some reflection is

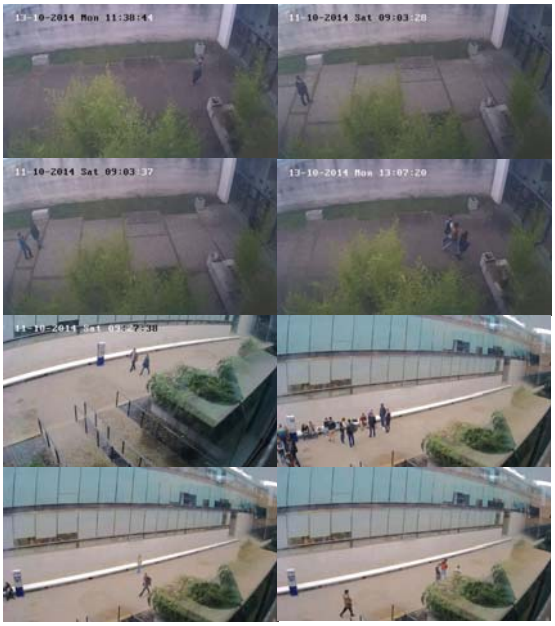

Figure 2: Examples of images from the collected videos (top four images are related with outdoor scenes, while the bottom four images are from indoor scenes).

captured), and far away from the human walking environment (some humans are captured with a low pixel definition); there are different weather conditions in the outdoor campus environment (e.g., rain and wind) and varying illumination in both indoor and outdoor areas due to different daytime recordings; there are clutter scenes in both areas due to the presence of trees and bushes; often, the human clothing includes colors that are very similar when compared with the background; and other uncontrolled conditions. The images from both indoor and outdoor scenes were used to test the human and star skeleton detection components (modules 1 and 2 of the proposed system). However, the final pedestrian prediction area (module 3) was tested only on the outdoor environment, since it was associated with a richer set of human trajectories and destination areas.

2.2. Intelligent System

The overall framework of the proposed intelligent system is shown in Figure 3. The whole system is composed of three main modules: human detection (module 1), star skeleton detection (module 2) and trajectory destination prediction (module 3). The intelligent system accepts a video as an input to the human detection module. This module detects all of the human blobs in the current video frames, performs background subtraction and output the respective blobs (detected objects). The output from module 1 (processed videos with detected blobs) serves as an input for the module 2. This subsystem works on further enhancing the detected blobs by performing shadow and highlight removal. Star skeleton is obtained from the detected blobs which is basically calculated by extracting the silhouette and finding peak points of the zero-crossing function. The peak points are connected to the center body of mass. The output from this module is a star skeleton that is further used by module 3 for prediction of destination trajectories. (Figure 10 shows some examples of

the detected star skeletons). Motion features for destination prediction system are extracted from this star skeleton and classification is performed using oversampling and undersampling with growing window validation. The final output from this module is the destination prediction, as provided by the prediction model. These modules are detailed in the next subsections. All system components were implemented in the Matlab system (Borse, 1996), except for the classification, which was implemented using the rminer library of the R tool (Cortez, 2010).

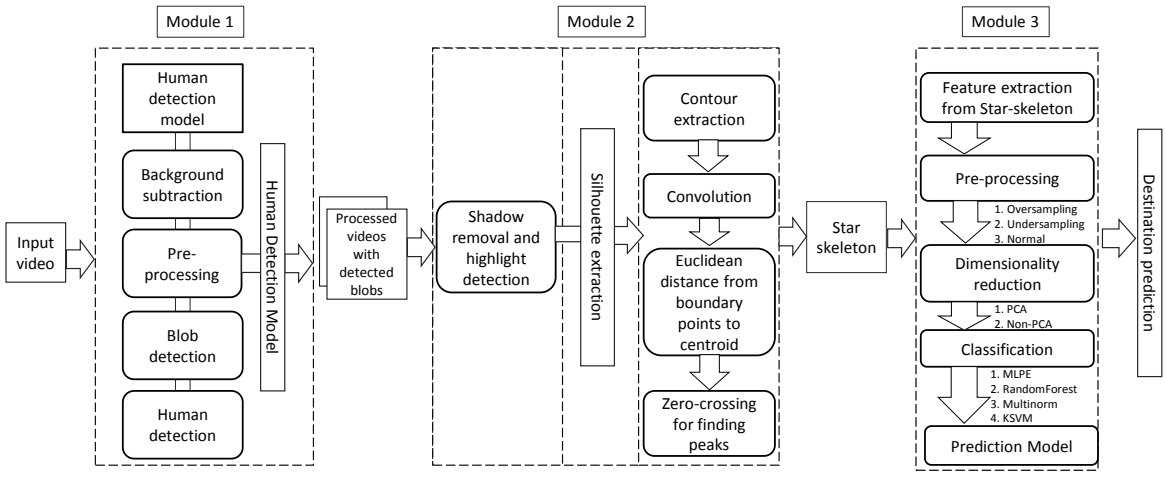

Figure 3: Overall framework of the system

2.2.1. Human detection

The field of human action recognition can be connected to many other disciplines that analyze human motion from videos. The recognition of basic human actions from monocular images and videos (e.g., sitting, walking, waving hands, jumping) is an important task in many computer vision applications, such as video content retrieval, surveillance and human computer interaction. Over the years, researchers have proposed many approaches for human action recognition. A detail survey is presented in (Afsar et al., 2015b).

The human detection of our system is detailed in (Afsar et al., 2017) and it will be briefly described here. It is based on segmentation method and blob detection. **Segmentation** is the process of partioning or dividing a digital image to look for objects of interest. **Blob detection** aims at detection regions in a digital image that differ from its surrounding pixels in terms of brightness or color. We tested two methods for image segmentation: usage of Gaussian mixture models (Kaewtrakulpong and Bowden, 2002) and the simpler background subtraction used in (Duque et al., 2007). The former method required more computational effort and also provided worst results. As exemplified in Figure 4, the method is not able to detect the legs separately, while the simpler image subtraction approach is. Therefore, we selected the simpler background subtraction, where a color image is compared against the background frame to identify if the pixels belong the background or moving object. This image differencing allows the definition of a foreground mask used by the blob detection algorithm, which groups pixels that most likely correspond to objects. **Mask image** is the

final image obtained after the subtraction of background image and current frame. After preliminary experiments, we set a minimum pixel area of 2000 pixels (e.g., 45×45, 60×34) for blob selection and that allows to reduce noisy non-human elements. The final result of this first module is a human detection model. It should be noted that our system only addresses individual human blobs and not crowds. This means that our system can track several pedestrians in the same video scene, but only if these pedestrians are not closer to each other. It means pedestrians that overlap (with no clear separating space) from the camera angle point of view (on a scale that is more close to 1/10 of a meter).

Figure 4: Example of segmentation results (first column denotes the original input frame; second column shows the Gaussian mixture model result; and third column the background subtraction result).

Regarding the background subtraction, it was done by using the same position of the bounding box obtained through blob detection, as both frames (current image and background image) are of the same size. Initially, mask image was obtained using Equation 1 while both images (current image and background image) are in Red-Green-Blue (color RGB model based on additive color primaries) color space.

$$M = |B - I| \tag{1}$$

In Equation 1, M represent the resultant "mask image", B denotes the "background image" and I represents the "current frame". The resultant mask M is converted into grayscale and then to binary using Otsu's thresholding (Otsu, 1975). The major drawback of this approach is when the clothing of a pedestrian is similar to the background, in such cases, the results are not good due to information loss. For example, Figure 5 shows the original input

image a, the absolute subtraction image b and the resulting binary mask c which contains two blobs related with the same person rather than one. For solving this problem we used

Figure 5: a) Original input image, b) result of absolute subtraction, c) binary image and d) result with shadow removal and highlight detection

Equation 2:

$$M'(x,y) = \begin{cases} 1 & \text{if } M(x,y) \geq \tau. \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where M' is the final mask, $M(x,y)$ is the pixel value of the mask at location x and y, and τ is a threshold. Several values for τ were tested (e.g., $\tau \in \{10, 20, 30\}$). The best results were achieved with with $\tau = 30$ (as shown in Figure 6) and thus this was the selected threshold value.

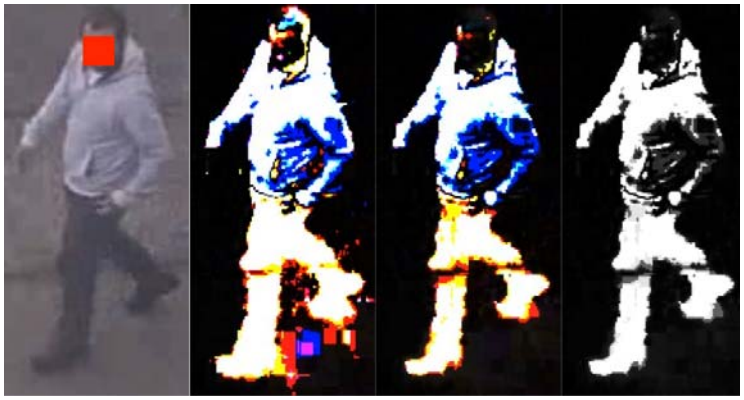

Figure 6: a) Original image, b) mask for τ=20, c) mask for τ=30 and d) grayscale mask for τ=30

2.2.2. Star skeleton detection

In order to obtain a correct human star skeleton, a perfect silhouette is necessary. In (Afsar et al., 2017) we explored several techniques to achieve this silhouette, such as thinning and zero-crossing. The best results were achieved using a shadow and highlight removal

combined with a zero-crossing star skeleton method, as detailed in the second module components of Figure 3.

To improve silhouette detection results, we used the shadow and highlight removal method proposed in (Duque et al., 2007) and that involves computing M' (Equation 2) using the Hue, Saturation, and Value (HSV) instead of RGB space, as exemplified in Figure 7. The resultant image SM (shadow mask) and LM (Highlight mask), represent the areas in the image where shadow and highlight are present and are computed as:

$$SM(x,y) = \begin{cases} 1 \text{ if } & \alpha \leq \frac{I^V(x,y)}{B^V(x,y)} \leq \beta \\ & \wedge \mid I^S(x,y) - B^S(x,y) \mid \leq \tau_S \\ & \wedge I^S(x,y) - B^S(x,y) \mid \leq \tau_H \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$SM(x,y) = \begin{cases} 1 \text{ if } & \frac{1}{\beta} \leq \frac{I^V(x,y)}{B^V(x,y)} \leq \frac{1}{\alpha} \\ & \wedge \mid I^S(x,y) - B^S(x,y) \mid \leq \tau_S \\ & \wedge I^S(x,y) - B^S(x,y) \mid \leq \tau_H \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In Equations 3 and 4, the value "1" denotes the pixels where there is a shadow or highlight. Also, the Hue, saturation and value components at coordinate *(x,y)* of input image I is represented by $I^H(x,y)$, $I^S(x,y)$ and $I^V(x,y)$ respectively. The same notion is applied for background image B. The main parameter is α and its value depends on the light source, radiance, and reflectance properties of the object in the scene. High reflective and high intensive light sources or irradiant objects can have low α values. For our dataset, α varies from 0.60 to 0.90. Decreasing the value below 0.60 causes information loss and increasing the value above 0.90, increases the noise in the final image.

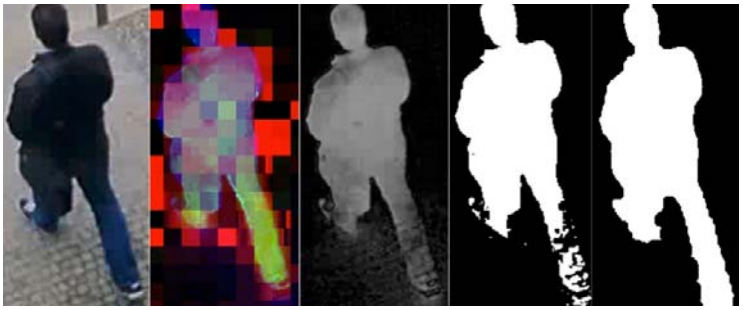

Figure 7: a) Original image, b) HSV image, c) *Value* component, d) binary image and e) result after shadow and highlight removal

We tested the algorithm with different α values, within the range {0.65,0.66,67,...,0.90} and the best results were achieved for α=0.70. After setting the value for α, we experimented different values for β, this prevents misclassification and the value varies in the data from 0.75 to 0.79. Since this is a less relevant parameter, we experimented distinct values but achieved the same results, thus this parameter was fixed to $\beta = 0.75$. The parameters τ_S

and τ_H are the maximal variation allowed for the saturation and hue components. We define τ_S as 15% of the digitizers saturation range. The variation of hue should not pass the 60 degrees. This value is obtained through the division of the hue range (360°) by the six colors (red, yellow, green, cyan, blue and magenta). The results obtained were satisfactory, as shown in Figures 5 d), 7 e) and 8.

Figure 8: Example of extracted human silhouettes using background subtraction combined with shadow and highlight removal

After achieving a human silhouette, the next step is to detect a star skeleton, which connects the extreme points (head, legs, hands) with the centroid (the body of mass). We note that human star skeleton is a key element in human action recognition systems (Fujiyoshi et al., 2004) and in our previous work it was quite useful for detecting human activities (e.g., walking, sitting) (Afsar et al., 2015a). In this paper, we compute a 5-star skeleton with the head, hands and legs points. This skeleton allows the estimation of movement features (used in by the third module) when considering point changes in two consecutive frames.

Figure 9 represents the overall procedure for obtaining a star skeleton. The original distance is plotted using the contour of the human silhouette. In order to smooth the human silhouette, a convolution method was applied. The peaks in this smooth distance function represents the points of star skeleton. Euclidean distances to the centroid were computed for all silhouette points, following a clock-wise order when processing the points. In the distance space, extreme points are associated with high peaks, which were detected using a zero crossing analysis over distance differences. Since often this method detects a large number of peaks, we defined a neighborhood threshold in the silhouette space and that was set to 40. Thus, all candidate points within the neighborhood range were aggregated by considering the median of such points, leading to a representative extreme point for that region. For instance, the left hand point, shown in the middle of Figure 10, was computed as the median of two neighbor peaks. Finally, for the construction of star skeleton, the detected points are connected with center body of mass. Also, in Figure 9, the points A, B, C, D, E and F represent the 5-star skeleton. Since both A and F are neighbor points, we assume the median of both and use it as a single representative point.

The full steps for the construction of a star skeleton algorithm are:

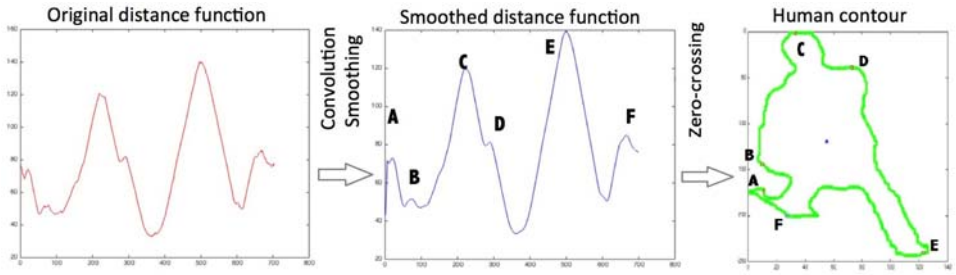

Figure 9: Process flow for the construction of a star skeleton

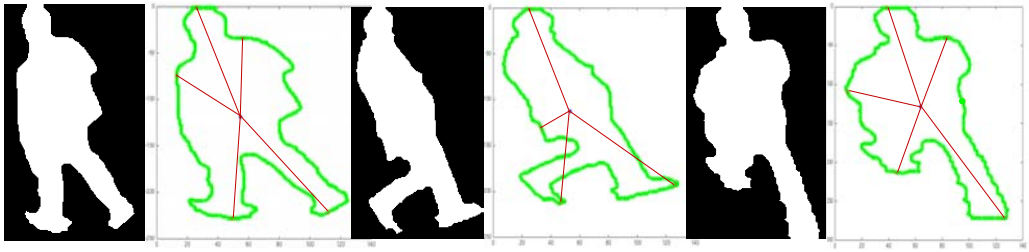

Figure 10: Some examples of star skeleton a) Binary Image b) our skeleton algorithm

1 Calculate the centroid of the contour of the input image (x_c, y_c).

$$x_c = \frac{1}{N_b} \sum_{i=1}^{N_b} x_i$$
$$y_c = \frac{1}{N_b} \sum_{i=1}^{N_b} y_i \tag{5}$$

where B_n are the number of boundary points and (x_c, y_c) denotes the centroid of the input contour.

2 Determine the distance d_i from each boundary point (x_i, y_i) to centroid (x_c, y_c).

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{6}$$

3 To remove the noise or unwanted peaks, the distance function is smoothed using convolution.

4 Find the local maximum by detecting zero crossing of the distance function differences $(d_{i+1} - d_i)$.

Figure 10 shows three result examples of the adopted star skeleton algorithm.

2.2.3. Pedestrian destination prediction

Once the star skeleton is achieved, we track its trajectory and extract its position and movement features, which are based on velocity measures (Kim et al., 2015). We assume that the video scene contains *a priori* set of human entry and exit areas. In particular, in this

paper, we assume the five entry and exit regions that are present in Figure 12. Nevertheless, we note that if such information is not available, the proposed system could automatically assign these entry and exit areas by analyzing a training set of trajectories. For each new human that is detected as entering and exiting scene (A, B, C, D or E), we create a new trajectory. Then, for all trajectory frames except the first two, we compute several measures for both x and y axis. These include the absolute center body mass position and the relative positions (distance to body of mass) of the head, hands and legs. Also, we compute the absolute velocity and acceleration of the body of mass and relative velocity and acceleration values for the five star peaks (head, hands and legs).

To assign human body part labels to the detected points, we have adopted several heuristics that were defined after analyzing a preliminary and small set of detected pedestrians. The heuristics are: head is the highest y value, legs are the lowest y values and hands are the middle y values. In most cases, the full star skeleton was detected. However, in some situations some start points are not detected (e.g., only one hand is detected in middle example of Figure 10). To solve this, we adopted these heuristics; when one hand is visible and the other is not, the second hand is set as the first one or it is assumed to be at the center body of mass Figure 11 a; when both hands are not detected, we assign them to the centroid Figure 11 c; when only one leg is detected, we assume that both legs are together; finally, if additional points are missing (e.g., head), we set them as the same position detected in the previous frame. The heuristics worked well in all cases analyzed. Figure 11 shows some examples of the labels detected.

Figure 11: Example of heuristics used a) Hand2 assumed to be at the center body of mass b) All of the points detected (leg1 and leg2 are assumed together as the legs are clsoed) c) Hand1 and Hand2 assumed to be at the center body of mass d) Hand1 and Hand2 assumed to be at the center body of mass.

Each classifier (Multinomial Logistic Regression, Multilayer Perceptron, Random Forest, Support Vector Machine) uses data known at a particular data frame, corresponding to current time t_c. A total of 37 inputs are used: one nominal entry point (A, B, C, D or E) and 12 numeric values for the position (centroid, head, hands and legs with respective x and y values), velocity and acceleration measures. As the target variable, it considers the nominal exit point: A, B, C, D and E (Figure 12). The analyzed exterior scene dataset includes a total of 171 videos related with 348 distinct pedestrian trajectories. Most pedestrian trajectories are short. The minimum pedestrian path required only $0.1s$, the median pedestrian trajectory time is $4.4s$, the average trajectory time is $5.3s$ and the maximum trajectory involves a total

of 62.4s. For each trajectory, there are several supervised learning examples, matching the 37 inputs of a particular trajectory frame with the target destination value. In total, the dataset contains 52,379 learning examples. The destination output class is also unbalanced, as some exit regions are much more common than others, namely: A - 36%, B - 4%, C - 2%, D - 48%, E - 5% and F - 5%.

Figure 12: Example of possible trajectories between the five A, B, C, D and E entry/exit regions.

We explore four classifiers in order to create the destination prediction models: Multinomial Logistic Regression (MLR), a Multilayer Perceptron (MLP) ensemble, Random Forest (RF) and Support Vector Machine (SVM). The classifiers were adopted with their default parameters, as defined by the rminer package of the R tool (Cortez, 2010). The MLR is the extension of the common logistic regression method for multiclass tasks. The MLP is a popular neural network where processing neurons are grouped into layers and connected by weighted links. The MLP hidden nodes were fixed to half of the input nodes and the MLP ensemble fits three networks, averaging their output responses into a single output. The RF uses an ensemble of 500 unpruned decision trees. Finally, the SVM is based in the standard regression model with a Gaussian kernel. The default SVM hyperparameters are $C = 1$, while the kernel parameter is set using a training data estimation heuristic defined in (Caputo et al., 2002).

Since our target class is unbalanced, we explore two balancing data methods (Menardi and Torelli, 2014) that tend to improve classification results for the minority classes: under sampling and over sampling. The formed method assumes all minority class examples and random replicates of other classes such that all classes are balanced. The latter method builds a larger balanced set by considering all majority class patterns plus a random over sampling of the minority classes. Both methods are only applied to training data and test data is kept with the original class distributions. We also explore a feature extraction method based on the well known Principal Component Analysis (PCA) method (Abdi and Williams, 2010). Using training data, we select the principal components that explain 95% of the variance, which allowed a reduction in the number of inputs from 37 (original inputs) to 25 (principal components). In total, each classifier (MLR, MLP, RF, SVM) is run using 6 data processing combinations, according to three balancing (no sampling, under sampling and over sampling) and two feature (all inputs and PCA extraction) setups.

2.3. Evaluation

We adopt the growing window evaluation scheme (Lopes et al., 2011), also known as incremental retraining evaluation, as presented in Figure 13. This scheme simulates a real usage of a classifier through time, in which the classifier is periodically updated. For instance, in the first day, the collected videos could be used to train (dataset initial size of W) a classifier, allowing it to produce predictions for the next day (test set of size T). At the end of the second day, the newly collected videos could be added to the training set (increasing the size of W), allowing the retraining of the classifier such that it can produce new predictions for the third day, and so on. Thus, under the growing window scheme, several iterations are executed, assuming a growing training set (as more data arrives) and a fixed test set size. In this work, the train test split is based on a timely ordering of the collected videos. In the first iteration, the oldest W videos are used to fit the classifier (training set), which then predicts the destinations for the frames of the next T videos (test set). In the next iteration, the test set is slided by adding data from a new video and discarding the oldest test set video data frames, which are merged into the training set, thus increasing its size. Similar iterations are then executed, until all videos are considered. In total, this scheme produces $U = L - (W + T)$ classifier updates (training and test iterations), where L is the data video length (the total number of videos in the dataset).

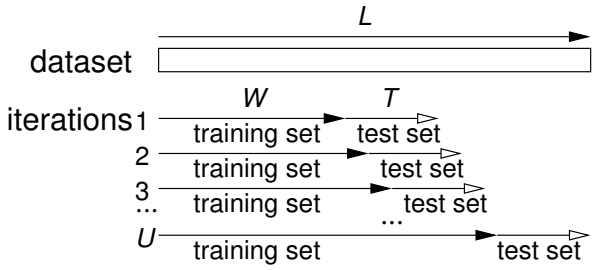

Figure 13: Schematic of the growing window evaluation.

To access the classification performance, we adopt the area under the curve (AUC) of the popular Receiver Operating Characteristic (ROC) curve analysis. A classifier can output a class probability ($p \in [0, 1]$) that according to a decision threshold D is interpreted as a positive class if $p > D$. Depending on the user selected threshold value (D), the classifier can produce a more sensitive (if a low D is selected) or specific response (high D value). The ROC curve shows the discrimination performance of the classifier for all possible D values, plotting the specificity (x-axis) versus the sensitivity (y-axis). In general, the quality of the AUC values are often interpreted as: up to 50% – poor (equivalent to a random classifier); 60% to 70% – good; 80% – very good; 90% – excellent; and 100% – perfect. Following the recommendations of Fawcett (2006), we compute the individual AUC class values, for each class, and also the global AUC, which weights the individual AUC values by each class prevalence in the data. Moreover, we execute a vertical aggregation method to estimate the median AUC curve, over all U runs, and its respective 95% confidence intervals, according to the Wilcoxon non parametric test. We also compute the ahead time (t_a, in seconds) in which the classifier predictions are computed (t_c) when compared to the time when the pedestrian

reaches the destination area (t_d). This time measure allows us to detail the quality of earlier predictions by plotting the AUC_a (AUC for a particular class and ahead time t_a, y-axis) versus the ahead time ($x-$axis) graphs. The measure also allows us to compute the quality time (t_q), the amount of ahead time in which the obtained AUC_a is higher than a quality threshold Q:

$$\begin{array}{rcl} t_a & = & t_d - t_c \\ t_Q & = & \max\{t_a : AUC_a \geq Q\} \end{array} \quad (7)$$

3. Results

All experiments were executed on a personal laptop running a Mac operating system. As previously explained, we only consider the exterior scene videos ($L = 171$) with five destination regions (A, B, C, D and E). We executed all steps of the proposed system (Figure 3) using a frame rate of 30 frames per second. During the classification stage, the initial training set size was defined to include around 2/3 of the videos ($W = 114$) and the test set was set with $T = 48$ videos, in a setup that results in $U = 14$ growing window runs.

Table 1 shows the obtained predictive results, in terms of the median AUC values (in %) for all classifiers and data processing setups. In general, excellent AUC values were achieved for class A (higher than 90%) and a very good discrimination was obtained for destination classes C and D (higher than 80%). Reasonable performances (AUC higher than 70%) were achieved for class E and some models, while the poorest discrimination was obtained for class B (ranging from 28.4% to 65.6%). Also, there seems to be no advantage in using PCA, as in general higher AUC values are achieved by the models that use all inputs. The discrimination performance seems more affected by the classifier type, with better overall results for the RF model, rather than the balancing sampling method.

For further analysis, we detail the classification results of two selected models. The first model is related with the best global AUC result and it corresponds to the RF using all inputs and the under sampling processing. For the second model, we wanted to select a classifier with a good class B AUC value (higher than 60%), since B is the most difficult class to discriminate. Since four SVM models provide such class B AUC values (bold values in Table 1), we selected one of these models as the representative of a good class B prediction, namely the SVM using all inputs and no sampling. This model was selected since it corresponds to one of the two preselected classifiers with the highest global AUC value (81.5%) and a better class B AUC value (65.4%) when compared with the all inputs and no sampling SVM classifier (64.5%).

For demonstrative purposes, we show the detailed results for the two selected models and three classes (A, B and D). The ROC curves of Figures 14 and 15 are plotted in terms of the vertical median and respective 95% confidence interval values. In the ROC graphs, the class destination performances are compared with a random classifier baseline (AUC=50%). The achieved ROC curves for the selected models confirm the excellent discrimination capability for class A (Figures 14 a and 15 a) and very good discrimination for class D (Figures 14 c and 15 c). For class B, the RF discrimination is weak (Figures 14 b and 15 b), similar to a random classifier, while the SVM performance is reasonable. Figures 14 and 15 also plot the ahead time versus AUC value for the same two selected models and three classes. The ahead time plots were computed using a time scale that goes up to $14s$, which includes 95%

Table 1: Discrimination performance on test data (median AUC values, in %; first model selected criterion values are underlined and the second model criteria values are in **bold**).

All	Undersampling				Oversampling				No sampling			
	MLR	MLP	RF	SVM	MLR	MLP	RF	SVM	MLR	MLP	RF	SVM
class A	88.9	94.6	96.3	93.7	89.7	95.7	95.5	89.5	92.9	94.9	95.5	92.4
class B	51.4	42.0	49.9	53.7	57.5	45.8	49.9	**65.4**$^{\diamond}$	59.9	41.7	46.6	**64.5**
class C	69.3	86.4	91.4	77.1	70.7	87.4	89.4	79.1	40.4	89.1	88.5	74.1
class D	80.2	81.8	84.1	78.2	79.4	83.2	84.5	80.2	75.6	82.9	83.8	78.8
class E	66.8	68.1	72.2	63.6	69.9	73.0	72.7	64.6	64.8	72.2	69.7	64.3
Global	80.8	84.3	86.9*	81.5	81.2	86.0	86.7	81.5	78.7	85.6	86.0	81.5
PCA	Undersampling				Oversampling				No sampling			
	MLR	MLP	RF	SVM	MLR	MLP	RF	SVM	MLR	MLP	RF	SVM
class A	94.2	94.0	94.3	93.1	93.2	94.9	93.2	88.7	95.2	94.3	94.2	91.6
class B	32.7	40.4	49.8	53.0	35.2	46.9	46.3	**65.6**	28.4	39.3	48.8	**65.3**
class C	85.7	86.9	84.0	79.7	84.2	89.0	84.3	81.1	81.4	87.6	84.0	76.6
class D	80.5	81.8	81.7	78.7	80.2	82.7	82.1	80.1	79.1	82.2	82.7	78.6
class E	71.2	69.6	69.0	64.5	71.9	70.9	69.7	64.9	68.5	69.9	66.7	64.0
Global	83.7	84.3	84.3	81.9	83.4	85.4	84.0	81.3	83.0	84.6	84.5	81.3

$\star$ - median 95% confidence interval within the range $[86.0, 87.8]$, statistically significant under a pairwise comparison against all other models except: all over sampling MLP and RF; all no sampling MLP and RF; and PCA MLP over sampling.

$\diamond$ - median 95% confidence interval within the range $[64.1, 65.0]$, statistically significant under a pairwise comparison against all other models except: all input no sampling SVM; PCA over sampling SVM; and PCA no sampling SVM.

of all pedestrian trajectories. For class A (Figures 14 d and 15 d), the ahead time plots show an almost stable discrimination performance for both RF and SVM selected models. More importantly, the ahead time graphs reveal that a high quality predictive performance (higher than $Q = 80\%$) is achieved for classes B (Figures 14 e and 15 e) and D (Figures 14 f and 15 f) (and the same selected models), although for a shorter advance time. For example, for class B and RF model, the time quality is t_q =6s when $Q = 80\%$ and $t_q = 7$ when $Q = 60\%$. The full time quality values, for all classes, two selected models and two quality values (very good – $Q = 80\%$, and reasonable – $Q = 60\%$) are shown in Table 2. These time quality results confirm the first model (all inputs, undersampling, RF) as the best model, since it provides better ahead quality times for all classes except B. And even for class B, the second selected model improvement is small (3 seconds for $Q = 80\%$ and 2 seconds for $Q = 60\%$). In effect, the selected RF provides very good ahead time quality values for four classes (A, B, C and D) and a reasonable ahead quality performance for class E.

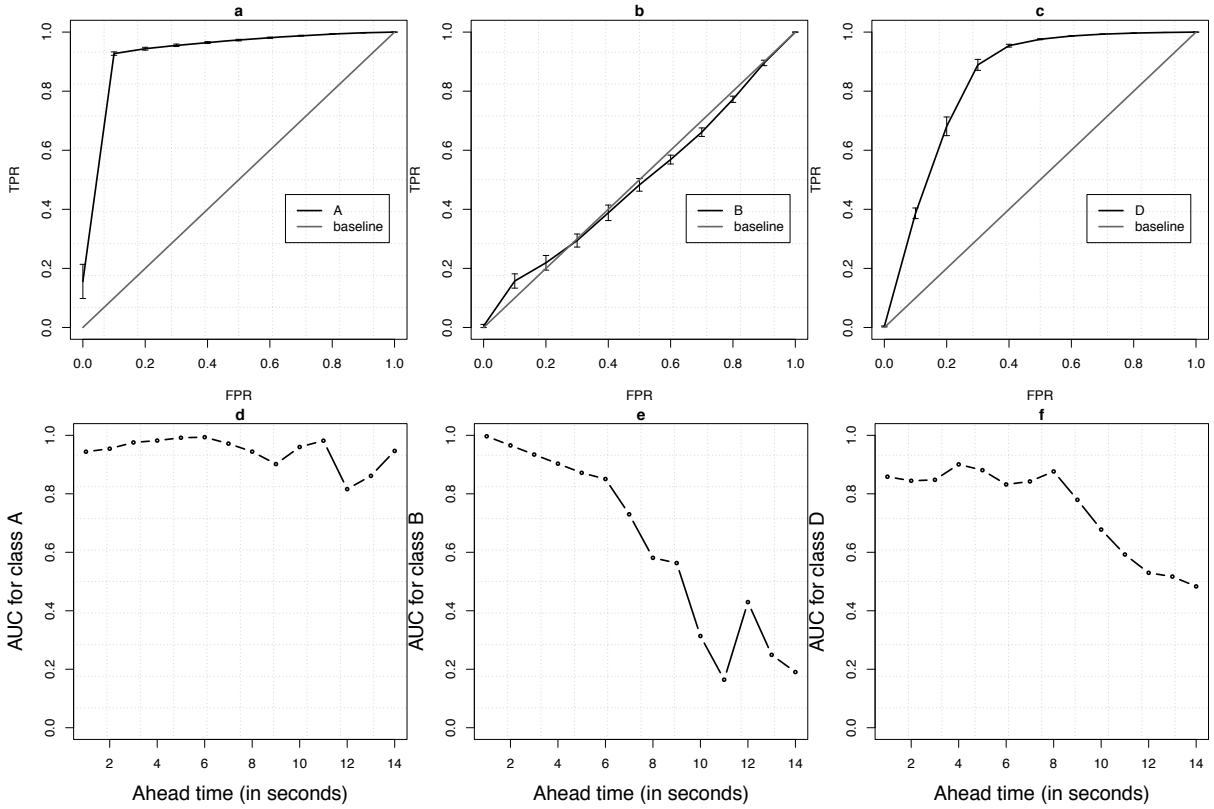

Figure 14: Overall ROC curve (top) and ahead time vs AUC graphs (bottom) for selected model 1 (all input under sampling RF) and destination classes A (left), B (middle) and D (right).

4. Conclusions

Nowadays, human daily activities are increasingly being recorded using digital cameras. In this paper, we present an intelligent system to predict the final destination area of pedestri-

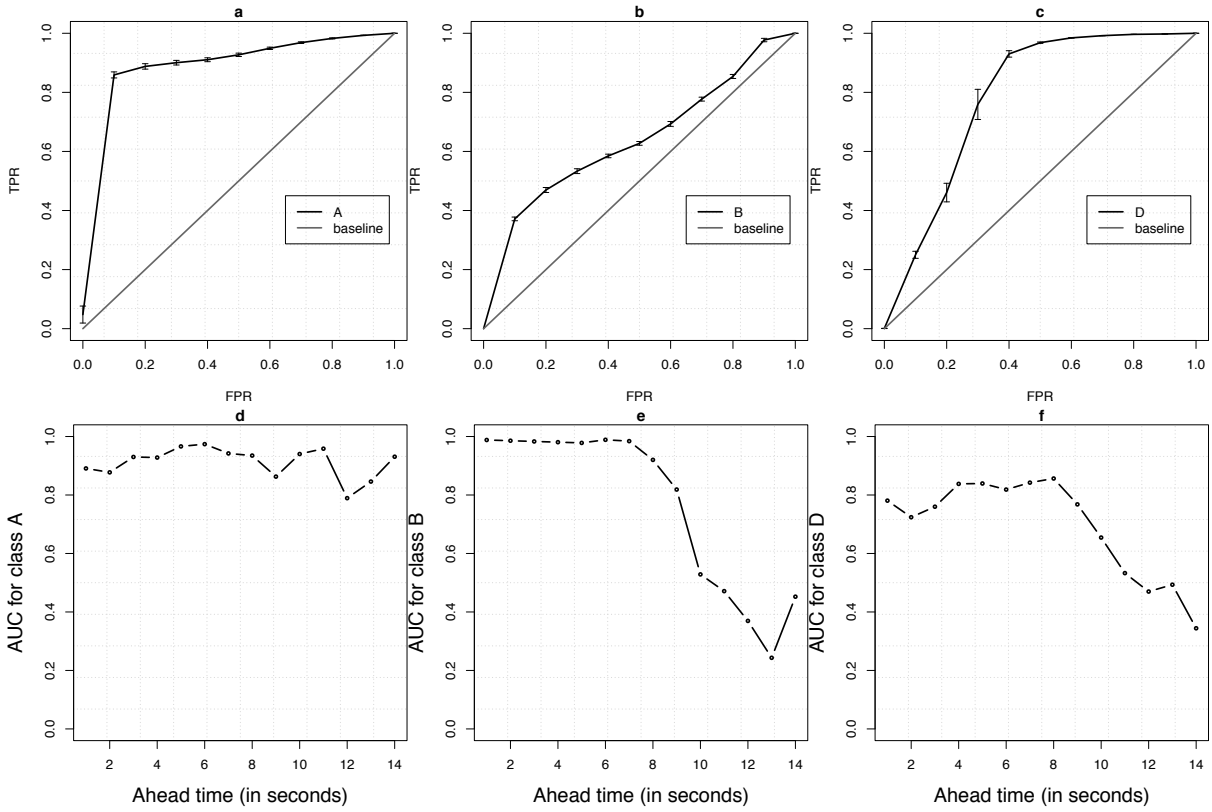

Figure 15: Overall ROC curve (top) and ahead time vs AUC graphs (bottom) for selected model 2 (all input over sampling SVM) and destination classes A (left), B (middle) and D (right).

ans moving freely in real-world environments. The proposed system adopts a passive collection of video, works directly with raw video data and extracts motion features for pedestrian destination prediction (position, velocity, acceleration) from automatically detected human skeletons (with positions of the body of mass, head, hands and legs). It includes three main modules: human blob detection – based on background subtraction; star skeleton detection – encompassing shadow removal and contour peak detection; and the final destination area prediction, based on preprocessing (dimensionality reduction and balancing sampling methods) and four classification methods: logistic regression, neural network, Random Forest (RF) and Support Vector Machine (SVM).

As a case study, we analyzed an exterior scene from a university campus and that includes five main destination areas (A, B, C, D and E). The collected dataset consisted of 348 pedestrian trajectories from 171 videos. The experimentation setup included a realistic growing window evaluation and the testing of four classifiers under six data processing combinations. The best results were achieved by the all inputs, under sampling and RF model. This model obtained the best global Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) analysis, which corresponds to a high quality class discrimination (median AUC of 87%). Moreover, the suggested model provided very good ahead time predictions for four of the classes (A, B, C and D) and a reasonable ahead discrimination performance for class

Table 2: Quality time (values in s).

	Q =80%		Q =60%	
class	Model 1	Model 2	Model 1	Model 2
A	14*	11	14*	14*
B	6	9	7	9
C	8	2	11	8
D	8	0	10	10
E	0	0	7	6

$*$ - at least 14s.

E.

The proposed intelligent system for destination area prediction from video achieved interesting results in the analyzed university campus case study. While tested in this case study, the system was designed for usage in a wide range of pedestrian walking real-world scenarios, since: it works directly from raw video data, captured using commonly used digital Closed-Circuit Television Cameras (CCTV); it assumes a passive collection of pedestrian paths (humans might not even know that the system exists); and it requires a minimum manual setting (e.g., initial adjust of the alfa parameter for shadow and highlight removal; optional setting of the destination areas of interest). As such, we believe the proposed system could be useful for other real-world application scenarios. For instance, it could be set to capture the interior of a large commercial store, anticipating when would a consumer approach a an important promotional product or cashier machine, triggering a temporal need for an employee (e.g., commercial vendor or cashier) at that position. Moreover, the same system could be used outside a factory with several near buildings and doors, antecipating the need for preparing more meals or waiting staff (e.g., when entering a canteen) or triggering a warning or security alarm (e.g., when approaching a very restricted zone).

In future work, we aim to test our system in these application environments. Moreover, we intend to enrich our system with additional features, such as scene obstacles (Kim et al., 2015) neighborhood motion (Fernando et al., 2017) and other body part motion capture (e.g., knees, limbs) attributes. In particular, motion capture technology, such as proposed in (Zhang and Shah, 2015; Girdhar et al., 2017), could be relevant for the detection of more advanced body parts.

Acknowledgments

This work is funded by the Portuguese Foundation for Science and Technology (FCT - Fundação para a Ciência e a Tecnologia) under research grant SFRH/BD/84939/2012.

References

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Afsar, P., Cortez, P., and Santos, H. (2015a). Automatic human action recognition from video using hidden markov model. In *Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on*, pages 105–109. IEEE.

Afsar, P., Cortez, P., and Santos, H. (2015b). Automatic visual detection of human behavior: A review from 2000 to 2014. *Expert Systems with Applications*, 42(20):6935–6956.

Afsar, P., Cortez, P., and Santos, H. (2017). Human skeleton detection from semi-constrained environment video. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP (5: VISAPP)*, volume 5: VISAPP, pages 384–389, Porto, Portugal.

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971.

Borse, G. J. (1996). *Numerical methods with MATLAB: A resource for scientists and engineers*. International Thomson Publishing.

Caputo, B., Sim, K., Furesjo, F., and Smola, A. (2002). Appearance-based object recognition using svms: which kernel should i use? In *Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision, Whistler*, volume 2002.

Cermeño, E., Pérez, A., and Sigüenza, J. A. (2018). Intelligent video surveillance beyond robust background modeling. *Expert Systems with Applications*, 91:138–149.

Cortez, P. (2010). Data mining with neural networks and support vector machines using the r/rminer tool. *Advances in data mining. Applications and theoretical aspects*, pages 572–583.

Cortez, P., Matos, L. M., Pereira, P. J., Santos, N., and Duque, D. (2016). Forecasting store foot traffic using facial recognition, time series and support vector machines. In Graña, M., López-Guede, J. M., Etxaniz, O., Herrero, Á., Quintián, H., and Corchado, E., editors, *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16 - San Sebastián, Spain, October 19th-21st, 2016, Proceedings*, volume 527 of *Advances in Intelligent Systems and Computing*, pages 267–276.

Duque, D., Santos, H., and Cortez, P. (2007). Prediction of abnormal behaviors for intelligent video surveillance systems. In *Computational Intelligence and Data Mining. CIDM 2007. IEEE Symposium on*, pages 362–367. IEEE 2007.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Fernando, T., Denman, S., Sridharan, S., and Fookes, C. (2017). Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *arXiv preprint arXiv:1702.05552*.

Fujiyoshi, H., Lipton, A. J., and Kanade, T. (2004). Real-time human motion analysis by image skeletonization. *IEICE TRANSACTIONS on Information and Systems*, 87(1):113–120.

Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., and Tran, D. (2017). Detect-and-track: Efficient pose estimation in videos. *arXiv preprint arXiv:1712.09184*.

Kaewtrakulpong, P. and Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer.

Kim, S., Guy, S. J., Liu, W., Wilkie, D., Lau, R. W., Lin, M. C., and Manocha, D. (2015). Brvo: Predicting pedestrian trajectories using velocity-space reasoning. *The International Journal of Robotics Research*, 34(2):201–217.

Kratz, L. and Nishino, K. (2012). Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):987–1002.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. *arXiv preprint arXiv:1704.04394*.

Lin, K., Chen, M., Deng, J., Hassan, M. M., and Fortino, G. (2016). Enhanced fingerprinting and trajectory prediction for iot localization in smart buildings. *IEEE Transactions on Automation Science and Engineering*, 13(3):1294–1307.

Lopes, C., Cortez, P., Sousa, P., Rocha, M., and Rio, M. (2011). Symbiotic filtering for spam email detection. *Expert Systems with Applications*, 38(8):9365–9372.

Luber, M., Stork, J. A., Tipaldi, G. D., and Arras, K. O. (2010). People tracking with human motion predictions from social forces. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 464–469. IEEE.

Mabrouk, A. B. and Zagrouba, E. (2017). Abnormal behavior recognition for intelligent video surveillance systems: a review. *Expert Systems with Applications*.

Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, pages 1–31.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.

Qiao, S., Shen, D., Wang, X., Han, N., and Zhu, W. (2015). A self-adaptive parameter selection trajectory prediction approach via hidden markov models. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):284–296.

Robicquet, A., Sadeghian, A., Alahi, A., and Savarese, S. (2016). Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer.

Sadeghian, A., Alahi, A., and Savarese, S. (2017). Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6.

Yamaguchi, K., Berg, A. C., Ortiz, L. E., and Berg, T. L. (2011). Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE.

Zhang, D. and Shah, M. (2015). Human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2012–2020.

Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016). Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975.